

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Absolvování individuální odborné praxe

Individual Professional Practice in the Company

Zadání bakalářské práce

Student:

Petr Tengler

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R025 Informatika a výpočetní technika

Téma:

Absolvování individuální odborné praxe
Individual Professional Practice in the Company

Jazyk vypracování:

čeština

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: NetDirect s.r.o.
2. Struktura závěrečné zprávy:
 - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
 - b) Seznam úkolů zadáných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
 - c) Zvolený postup řešení zadáných úkolů.
 - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
 - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
 - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

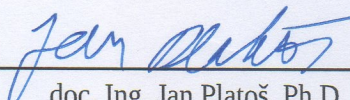
Vedoucí bakalářské práce: **doc. Mgr. Jiří Dvorský, Ph.D.**

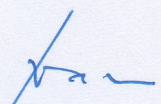
Konzultant bakalářské práce: Bc. Magdaléna Velická

Datum zadání: 01.09.2017

Datum odevzdání: 30.04.2018




doc. Ing. Jan Platoš, Ph.D.
vedoucí katedry


prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne: 25. června 2018

.....
Tengler

Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava.

Dne: 25. června 2018

Ing. Michal Fabík

Rád bych poděkoval doc. Mgr. Jiřímu Dvorskému, Ph.D. za konzultace k této práci a kolektivu firmy NetDirect s.r.o., který mi umožnil vykonávat bakalářskou praxi a za jejich odbornou pomoc při vytváření této bakalářské práce.

Abstrakt

V bakalářské práci je popsána individuální odborná praxe, kterou jsem vykonal ve firmě NetDirect s.r.o. v Ostravě. Tato firma se zabývá výrobou a správou e-shopů. Během praxe jsem působil na pozici programátora v servisním oddělení. Bakalářská práce detailně rozebírá pracovní postupy, které jsem využil k řešení jednotlivých úkolů zadaných buď klientem, nebo firmou. V práci jsou také popsány znalosti a dovednosti využité či získané v odborné praxi.

Klíčová slova: Odborná praxe, NetDirect, C#, VBScript, JavaScript, jQuery, XSLT, T-SQL

Abstract

In the bachelor thesis is described individual professional practise which I performed in company NetDirect s.r.o. in Ostrava. The company is focusing on e-shops production and administration. During professional practise I was employed as programmer in servis department. The bachelor thesis is analysing working procedures which I used for solving particular tasks which was assigned either by clients or by the company. The bachelor thesis is also targeted to describe knowledge and skills used or acquired in the professional practise.

Key Words: Professional practise, NetDirect, C#, VBScript, JavaScript, jQuery, XSLT, T-SQL

Obsah

Seznam použitých zkratek a symbolů	8
Seznam obrázků	9
Seznam výpisů zdrojového kódu	10
1 Úvod	11
2 Firma NetDirect	12
2.1 O firmě	12
2.2 Mé působení	12
3 Zadané úkoly	13
3.1 Vytvoření nového příznaku	13
3.2 Špatně zobrazený text	14
3.3 Rozbalující se postranní menu	15
3.4 Změna hesla	18
3.5 Chyba s bannery v administraci	20
3.6 Nefunkční stránka	21
3.7 Notifikace	21
4 Využité znalosti a dovednosti	27
5 Chybějící znalosti a dovednosti	28
6 Závěr	29
Literatura	30

Seznam použitých zkratek a symbolů

AJAX	– Asynchronous JavaScript and XML
API	– Application Programming Interface
ASP	– Active Server Pages
CSS	– Cascading Style Sheets
DB	– Database
HTML	– Hyper Text Markup Language
ID	– Identification
SQL	– Structured Query Language
T-SQL	– Transact-SQL
URL	– Uniform Resource Locator
VAPID	– Voluntary Application Server Identification for Web Push
XML	– Extensible Markup Language
XSLT	– Extensible Stylesheet Language Transformations

Seznam obrázků

1	Značka s písmenem „T“ u produktu je označení pro nový příznak	14
2	Postranní menu před a po úpravě	18
3	Přijatá notifikace v prohlížeči Chrome	25
4	Správa notifikací v administraci	26

Seznam výpisů zdrojového kódu

1	Jednoduché přidání prvku k rozbalování menu	16
2	Funkce pro otáčení rozbalovacími ikonami	16
3	Funkce pro vygenerování náhodného hexadecimálního řetězce	19
4	Funkce pro kontrolu nově příchozích notifikací	22
5	Hledání shody odkazu s otevřenými panely	23

1 Úvod

Využil jsem možnost absolvovat individuální odbornou praxi za účelem převést nabyté znalosti ze školy do reálných projektů a také, abych zjistil více o tom, jak takový projekt ovlivňuje práce v týmu. Zaškolení v průběhu odborné praxe může také pomoci v budoucích pracovních příležitostech. Vybral jsem si firmu NetDirect s.r.o., která mne přijala na oddělení ShopCentriku – servis, do týmu několika lidí na pozici programátora, který má přímý kontakt s koncovým uživatelem a musí se orientovat od databáze až po uživatelské rozhraní.

V dalších kapitolách stručně představím firmu a její zaměření, úkoly, které jsem musel v průběhu praxe vyřešit a na závěr celkově zhodnotím praxi.

2 Firma NetDirect

2.1 O firmě

Firma NetDirect s.r.o. se zabývá převážně vytvářením a správou e-shopů na mezinárodní úrovni. Firma se dále dělí na hlavní dvě oddělení FastCentrik a ShopCentrik. FastCentrik je krabicové řešení, které je levné a hodí se pro každého, kdo si chce pořídit menší e-shop a ShopCentrik, který vytváří e-shopy šité přesně na míru daného klienta a jeho požadavků a také umožňuje individuální změny a vylepšení během celého života e-shopu [1]. ShopCentrik se dále dělí na oddělení výroby, které vytváří nové e-shopy a stará se o ně i po určitou dobu po implementaci, a na oddělení servisu, které tyto projekty následně přebírá a stará se o ně až do konce jejich existence.

NetDirect s.r.o. pracuje na technologiích od společnosti Microsoft:

- Microsoft Visual Studio – C#, VBScript
- SQL Server Management Studio – T-SQL

2.2 Mé působení

V průběhu své odborné praxe jsem působil v ShopCentrik na oddělení servisu, který se z převážné části věnuje individuálním požadavkům klientů od oprav, přes úpravy, až po implementaci nových technologických vylepšení, které v původně vyrobeném projektu nebyly. Tyto požadavky se řeší pomocí ticketů, který si klient založí pro každý úkol zvlášť. Kromě tohoto systému úkolů jsem dostával také úkoly přímo od firmy, kdy výsledky těchto úkolů bylo možné klientům aktivně nabídnout. Při své práci jsem využíval technologie C# [2], T-SQL [3], VBScript [4], JavaScript [5], XSLT [6], XML [7], HTML [8], CSS [9], jQuery [10], AJAX [11] a další.

3 Zadané úkoly

3.1 Vytvoření nového příznaku

3.1.1 Zadání

Mým úkolem bylo přidat ke zboží nový příznak a zároveň vytvořit URL adresu, na které se bude nalézat vyfiltrované zboží z katalogu produktů právě podle tohoto příznaku.

3.1.2 Analýza

Při procházení tohoto e-shopu jsem našel naimplementované příznaky pro akční zboží, doporučené a novinky, podle kterých jsem se dále v kódu orientoval.

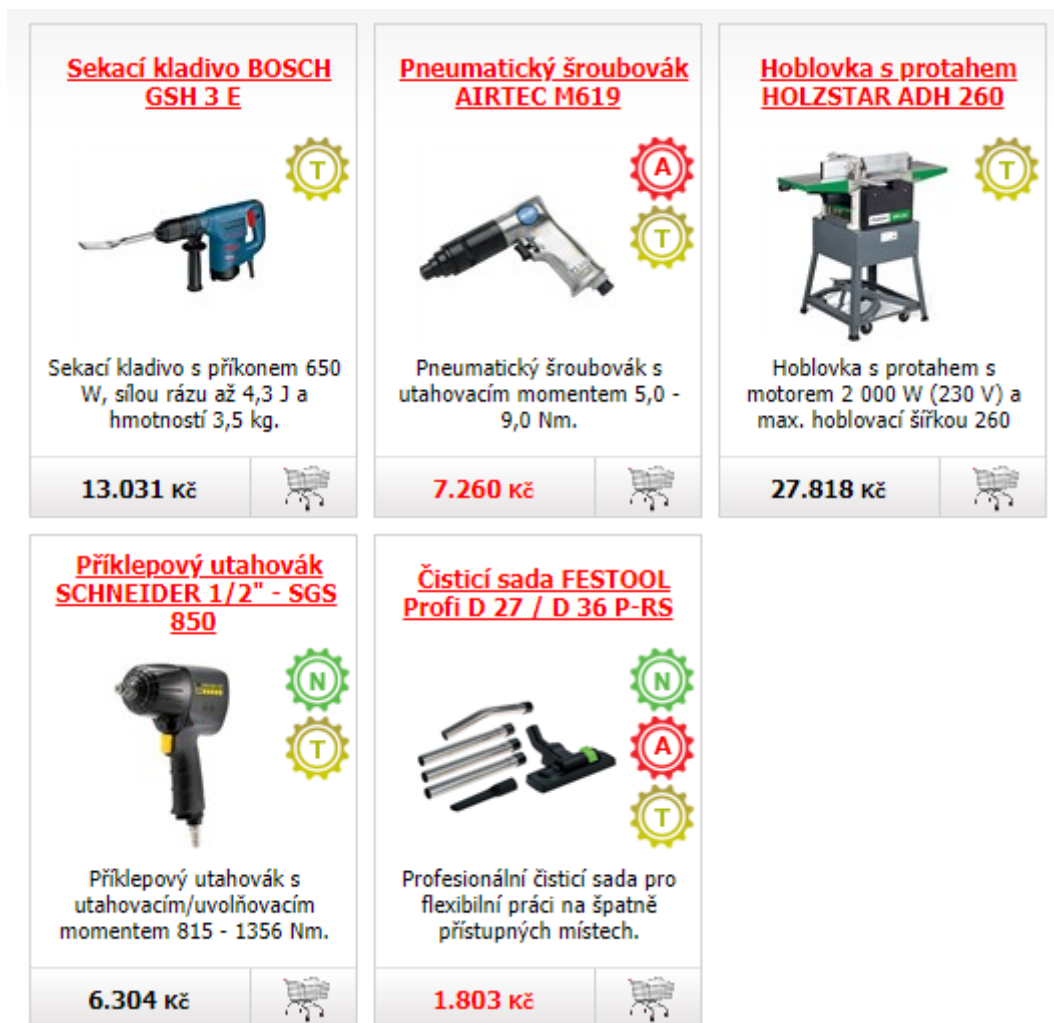
3.1.3 Řešení

Nejprve jsem v databázi rozšířil tabulku se zbožím o svůj nový příznak a do některých náhodných položek jej označil. Pomocí nástroje SQL Search, jehož hlavním úkolem je vyhledávat fragmenty SQL v tabulkách, pohledech, procedurách, funkcích a dalších [12], jsem si vyhledal procedury, do kterých zasahuje akční, doporučené zboží a novinky, do těchto procedur jsem doimplementoval svůj příznak, popřípadě jsem vytvořil novou proceduru. Vyzkoušel jsem si zde také práci s nástrojem ApexSQL Search, který vyhledává mimo jiné i v datech. Přidal jsem také nový záznam do souboru .htaccess, konfiguračnímu souboru, ve kterém se pomocí regulárních výrazů nastaví, co se má po zavolání příslušné URL adresy vykonat, v mém případě na který .aspx soubor se má přesměrovat. V XSLT šabloně potom zbývalo převést zdrojová data z XML do HTML a sestavit grafické rozhraní konečnému výpisu. Tento výpis bylo dále potřeba v souboru CSS dostylovat. Úkol byl řešen jako samostatný modul, který může klient spravovat přes administraci. Vzhledem k tomu, že už naimplementované příznaky měly u daného produktu svůj obrázek („razítko“), aby byl na první pohled u zboží viditelný i můj příznak, upravil jsem v grafickém programu obrázek od akčního produktu a přidal jej k produktům s mým příznakem. Na obr. 1 je vyfiltrováno zboží podle nového příznaku.

3.1.4 Zhodnocení

Použité technologie: T-SQL, XSLT, C#, XML

Časová náročnost: 4 dny



Obrázek 1: Značka s písmenem „T“ u produktu je označení pro nový příznak

3.2 Špatně zobrazený text

3.2.1 Zadání

Klient nahlásil problém se špatným zobrazením některých textů u jednoho produktu. Podle přiloženého obrázku byl poslední řádek příslušného textu zhruba ze spodní čtvrtiny oříznutý.

3.2.2 Analýza

Když jsem se snažil objevit chybu, měl jsem nejprve problém ji najít, jelikož jsem používal prohlížeč Chrome. Po otestování dalších prohlížečů jsem objevil, že každý prohlížeč zobrazuje daný text trochu jinak, ale správně pouze prohlížeč Chrome. A také chyba, která měla být u jednoho produktu, se objevovala na mnohem více místech, tam kde byl text příliš dlouhý –

3 řádky a případně více. Od kolegů jsem zjistil, že původně byl daný e-shop stavěn pouze na řádky dva.

3.2.3 Řešení

Začal jsem zvětšovat velikost HTML elementu, který obsahoval oříznutý text, ale brzy jsem přišel na to, že toto řešení funguje vždy právě pro jeden prohlížeč a ostatní to rozladí, a protože produkt s nejdelším textem měl řádků asi 6, nemohl jsem nikdy zvětšit prvek dostatečně, abych se zároveň vešel do buňky katalogu produktů. Při studování problematiky jsem objevil, že prohlížeče používají jiné velikosti řádku, ale dají se sjednotit v CSS pomocí vlastnosti line-height. Vyhledal jsem si, jakou velikost line-height používá prohlížeč Chrome a postupně ji pak nastavil v souboru CSS všem potřebným HTML elementům. Při závěrečném testování před odevzdáním klientovi jsem našel ještě asi 1 pixel chybějící z posledního řádku, který se projevoval nevykreslením podtržení, když se v textu objevilo nějaké velké písmeno. Při zvětšování HTML elementu, ve kterém byl text umístěn, se výsledek nedostavil, paradoxně pomohlo až drobné zmenšení, ve kterém se konečně text správně naformátoval. Na obr. 1 je možné vidět již opravené texty nad i pod ilustracemi produktů.

3.2.4 Zhodnocení

Použité technologie: CSS, HTML

Časová náročnost: 2 dny

3.3 Rozbalující se postranní menu

3.3.1 Zadání

Cílem tohoto úkolu bylo upravit postranní menu, aby znovu nenačítalo stránku, když uživatel chce rozbalit kategorii ve stromu postranního menu, ale zároveň zachovat stávající funkci – možnost přímo otevřít kategorii.

3.3.2 Analýza

V položkách postranního menu se sice nalézaly ikonky, které na první pohled navozovaly dojem rozbalujícího se menu, ale fungovaly pouze jako grafický doplněk, tedy stejně jako kliknutí na kategorii, a dokonce byly nastaveny do pozadí jednotlivých položek, proto nebylo možné jim jednoduše nastavit odlišnou funkci od zbývajících textové části.

3.3.3 Řešení

Pro nastavení, aby stromová struktura v postranním menu si z databáze vzala všechny položky a ne pouze aktivní, bylo potřeba v šabloně XSLT zmodifikovat příslušné funkce. Pro docílení

efektu rozbalování jsem zvolil přidání druhého HTML elementu do odrážek stromové struktury, po jehož zakliknutí se menu otevře, či zavře. Jako nejjednodušší řešení, které by zároveň odlišovalo, kdy je kategorie otevřená, či ne, se ukázalo vložit do nového HTML elementu znak „+“ pro nerozbalené odrážky a pomocí jednoduchého JavaScriptu jej po kliknutí změnit na znak „–“ a kategorii rozbalit nebo obráceně.

```
<xsl:if test="@nChildren>0">
  <span class="rozbal" onclick="rozbalit('catUl{$currId}')">
    <span class="pluscatUl{$currId}" >
      +
    </span>
    <span class="minuscatUl{$currId}" style="display:none" >
      -
    </span>
  </span>
</xsl:if>
```

Výpis 1: Jednoduché přidání prvku k rozbalování menu

Tato varianta se mi ve výsledku vzhledem k už existujícímu „tlačítku“ v pozadí a celkovému designu e-shopu nelíbila, proto jsem raději vytáhl z odrážky pozadí, ořízl obrázek a vložil jej do samostatného HTML elementu vedle textu do příslušné odrážky, zároveň bylo nutné opravit rozladěné velikosti odrážek a jejich formát. Rozhodl jsem se strohé rozbalování oživit, a proto jsem přidal jednak do funkce pro rozbalování postupný pohyb, a jednak také jQuery funkci, která s animací otáčela ikonami pro rozbalování menu.

```
$(document).ready(function () {
  $('.openingImage').each(function () {
    elem = this;
    elem.notRotated = 1;
  });

  $('.openedImage').each(function () {
    elem = this;
    $(elem).css({ 'transform': 'rotate(' + 90 + 'deg)' });
    elem.notRotated = 0;
  });

  $('.openingImage, .openedImage').click(function () {
    elem = this;
    $({ rotation: 90 * !elem.notRotated }).animate({ rotation: 90 * elem.
      notRotated }, {
```

```

        duration: 400,
        step: function (now) {
            $(elem).css({ 'transform': 'rotate(' + now + 'deg)' });
        }
    });
    elem.notRotated = !elem.notRotated;
});
});

```

Výpis 2: Funkce pro otáčení rozbalovacími ikonami

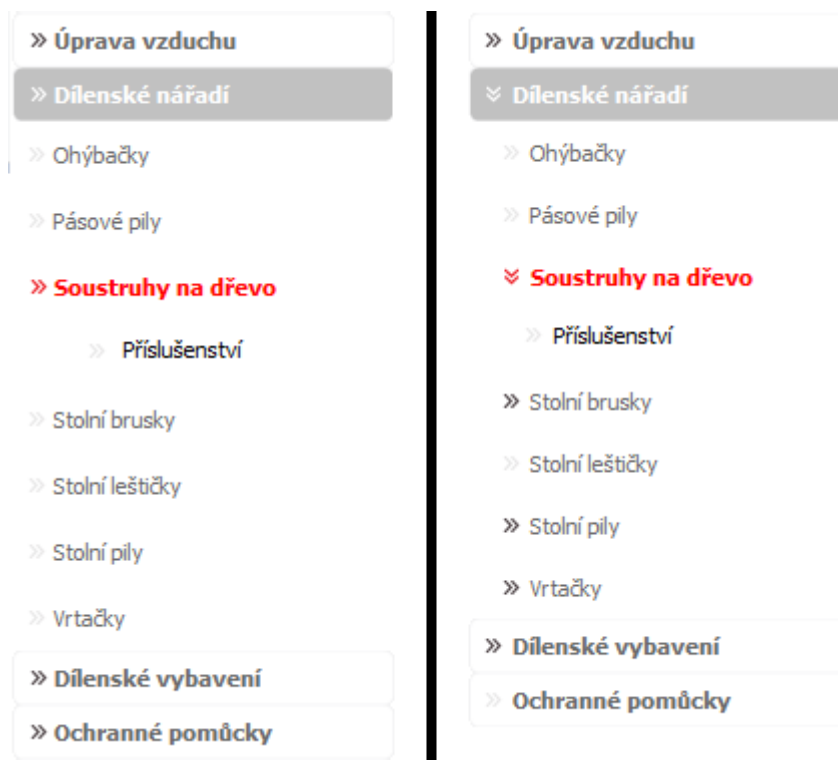
Jelikož klient využívá pro různé úrovně stromu a aktivní kategorie malinko odlišné ikony, musel jsem také pro zachování vzhledu stránky postup s ikonami částečně opakovat a rozlišovat je, tento vzhled jsem trochu pozměnil, aby zákazník tohoto e-shopu věděl, které kategorie je možné rozbalovat. Při testování výsledku se objevila chyba, kdy se kategorie záhadným způsobem stále zanořovaly, místo aby se skládaly za sebe, jestliže se uživatel nepohyboval po hlavní stránce, ale byl zanořený do nějaké větve tohoto stromu – nějaká kategorie byla aktivní.

Zaměřil jsem se tedy na položku, jenž získala atribut „aktivní“ a v příslušné XSLT šabloně po řádcích sledoval, kde kódem prochází, a upravoval jednotlivé podmínky větvení či cyklů, které by mohly souviset se špatným výpisem. Výsledky těchto úprav byly někdy i vizuálně přijatelné, ale po otestování se vždy našla nějaká jiná nově vygenerovaná omezení. Pomocí kolegovy návodu jsem zprovoznil debug v XSLT šabloně, a přestože mi Visual Studio neukázalo obsah proměnných ani se modul stránky nenačítal postupně, nakonec jsem našel cyklus, který fungoval jinak pro aktivní a neaktivní kategorii, ovšem nepovedlo se mi zjistit, proč se choval. Chyba se nakonec vytratila, až když se z databáze zakázalo posílat ID aktivní kategorie. Naštěstí proměnná v XSLT šabloně si udržovala ID všech rodičovských kategorií od právě načtené, nalezení ID kategorie v tomto řetězci fungovalo stejně jako atribut „aktivní“ a díky tomu jsem mohl znovu přiřadit správné ikony jednotlivým položkám menu.

3.3.4 Zhodnocení

Použité technologie: XSLT, CSS, JavaScript, jQuery, HTML, T-SQL, XML

Časová náročnost: 8 dní



Obrázek 2: Postranní menu před a po úpravě

3.4 Změna hesla

3.4.1 Zadání

Klient požádal o zlepšení zabezpečení v případě, že si registrovaný uživatel na klientově e-shopu zapomene své heslo k účtu.

3.4.2 Analýza

Vytvořil jsem si účet na webových stránkách klienta a vyzkoušel si nasimulovat průběh obnovy zapomenutého hesla. Uživateli stačí akorát zadat email, a pokud se v databázi najde, je mu obratem vygenerováno nové heslo s dočasnou platností, které je na zadaný email posláno společně s přihlašovacím jménem. Heslo je velmi citlivý údaj, se kterým není možné pracovat jako s běžně viditelným textem. Heslo musí být neustále šifrované a i práce s ním, tedy porovnávání shody, probíhá pouze v šifrovaném stavu, v tomto případě se používá algoritmus MD5. Dočasná platnost nově vygenerovaného hesla je jedno z řešení pro zapomenuté heslo. U tohoto kroku jsem postřehl, že se vlastně přihlašuji pomocí přihlašovacího jména nikoli emailu, proto jsem si vyzkoušel založit pod stejným emailem další účet a „zapomenout“ heslo. Úspěšně. Sice se nové heslo přepsalo ke všem účtům, ale byly všechny účty vypsané pěkně za sebou. Usoudil jsem, že ač mi z neznámého důvodu se na těchto starých webových stránkách s touto možností počítá.

3.4.3 Řešení

Nejprve jsem oddělil dočasná hesla a nechal je generovat ke každému účtu zvlášť. Kolega mi poradil a ukázal, jak by měla změna zapomenutého hesla ve výsledku fungovat. Vygenerované by nemělo být nové heslo, ale dočasný odkaz na stránku kde si změni heslo přímo uživatel. V databázi jsem si proto vytvořil funkci, která mi generovala náhodný řetězec hexadecimálních znaků, jedinečnosti tohoto řetězce jsem docílil porovnáním s již vygenerovanými řetězci.

```
DECLARE @index int
DECLARE @digit char(1)
DECLARE @seq varchar(16) = '0123456789abcdef'
DECLARE @length int = 20
DECLARE @result nvarchar(100)
DECLARE @sPassChangeCode nvarchar(100) = ''

WHILE @sPassChangeCode = ''
BEGIN
    SET @result = ''
    SET @index = 0

    WHILE @index < @length
    BEGIN
        SET @digit = cast(SUBSTRING(@seq, cast(ABS(CHECKSUM(NEWID()))%16+1 as int
        ), 1) as char)
        SET @result = @result + @digit
        SET @index = @index + 1
    END

    IF(not exists(SELECT * FROM tblClientContact WHERE sPassChangeCode = @result
    ))
        SET @sPassChangeCode = @result
END
```

Výpis 3: Funkce pro vygenerování náhodného hexadecimálního řetězce

Řetězec se z databáze smaže až při změně hesla, ale po 15 minutách se stává propadlým a uživatel se zapomenutým heslem by potřeboval vygenerovat tento řetězec znovu. V případě potřeby bylo možné tyto pravidla pro generování zpřísnit. Tento řetězec a jeho časové razítko bylo samozřejmě potřeba zanést k tabulce se zákazníky. V tomto okamžiku jsem si nebyl jistý, jak pokračovat dále, aby URL adresa, sjednocená s pokaždé jiným řetězcem znaků, přesměrovala právě na jednu webovou stránku, k tomu jsem si později dohledal regulární výraz do přesmě-

rovávacího souboru .htaccess. Po byt' zdařilém výsledku tohoto úkolu jsem musel přepracovat řešení jiného e-shopu, kde byla už tato úprava dříve zhotovená, aby se s e-shopy lépe pracovalo jako kolektivu a neměl každý unikátní řešení pro jednu a tu samou věc.

3.4.4 Zhodnocení

Použité technologie: C#, T-SQL, XSLT, XML

Časová náročnost: 6 dní

3.5 Chyba s bannery v administraci

3.5.1 Zadání

Klient nahlásil chybu, kdy nemohl manipulovat s bannery přes administraci.

3.5.2 Analýza

Přihlásil jsem se na administraci e-shopu daného klienta a testoval, co přesně klientovi nefunguje. Při výběru bannerů se mi povedlo k nim dostat, ale ne vždy, objevil jsem, že chybný výsledek se zobrazuje, pouze když vyhledávám všechny položky bez filtru na pozici banneru. Přestože chybová hláška ukazovala na přetypování času, ze začátku jsem na ni nereagoval, protože mi nedávalo smysl, že při použití filtru na pozici banneru tento problém nenastává. Navíc vyhledávač byl vybaven funkcí na vyhledávání podle času zvlášť a jeho funkce se s filtrem na pozici zdála v pořádku a bez něj vyhledávání nefungovalo.

3.5.3 Řešení

Po prohledávání příslušného .ASP souboru, jsem žádnou chybu spojenou s filtrováním nemohl najít. Vrátil jsem se proto zpět k chybové hlášce a vyhledal si kýžené řádky. Nechal jsem si vypisovat vstupní data těchto řádků na obrazovku a v případě, že se v řetězci objevil nějaký čas, stránka nefungovala správně. Vzhledem k tomu, že účelem těchto řádků bylo jen změnit formát času, vymazal jsem VBScript funkci, která přetypovávala řetězec na čas, protože jej nedokázala správně přečíst, a ponechal jsem formát času řetězce. Při ověřování funkčnosti jsem zjistil, že časové razítko obsahuje velmi málo bannerů a snad jen ve dvou kategoriích, na které jsem v původní nepřesné analýze zrovna nenarazil.

3.5.4 Zhodnocení

Použité technologie: VBScript

Časová náročnost: 1 den

3.6 Nefunkční stránka

3.6.1 Zadání

Klient požádal o vyřešení vážného problému, při kterém mu přestala fungovat právě jedna webová stránka na jeho e-shopu.

3.6.2 Analýza

Po projití všech možných kódů jsem usoudil, že by stránka měla fungovat správně, jak má. Zaměřil jsem se na otázky proč právě tato stránka nebo proč zrovna teď, když nikdo na stránkách nic neupravoval. Vzpomněl jsem si, že před pár dny, před obdržáním tohoto ticketu, se v rámci firemních projektů zvyšovala bezpečnostní opatření proti SQL injection a klient měl v názvu této stránky zakomponované klíčové slovo tohoto jazyka. Klient byl proto požádán o změnu názvu v nefunkční webové stránce, v případě jeho nesouhlasu by musel na vlastní nebezpečí požádat o výjimku z bezpečnostního pravidla pro tento případ u administrátorů.

3.6.3 Zhodnocení

Použité technologie: C#, T-SQL

Časová náročnost: 1 den

3.7 Notifikace

3.7.1 Zadání

Mým úkolem bylo vytvořit funkční WebPush notifikace, včetně administrace, ve které bude klient moci s nimi, dle svého uvážení, pracovat. Cílem tohoto úkolu byla také možnost toto řešení dále přenášet na další e-shopy.

3.7.2 Analýza

Studováním problematiky jsem narazil na několik možností řešení, kdy většina využívá službu třetí strany, neplacené ale i zpoplatněném, s podporou různých možností a rozdílně podporované prohlížeči.

3.7.3 Řešení

Nejprve jsem se pokusil řešit tento úkol bez služby třetích stran a využít k tomu klasické notifikace, které oproti Push notifikacím nejsou odesílány z backendu. V prvním kroku jsem si navrhnul administraci, která jednak umožňovala vytvářet nové notifikace, a jednak dokázala podle různých klíčů vyhledávat mezi odeslanými notifikacemi, k tomuto jsem samozřejmě potřeboval naimplementovat do databáze procedury, které s touto administrací komunikují a spravují

jednotlivé notifikace. Tomuto typu notifikací jsem naimplementoval základní možnosti – nadpis, obsah a URL odkaz po kliknutí. Pro zobrazení notifikace uživateli jsem využil technologie AJAX, kterou jsem vyvolával co určitou dobu (1 minuta) a vždy proběhla kontrola, jestli se v databázi neobjevily nové notifikace, které by se měly odeslat. Aby se po aktualizaci stránky nebo při prostém pohybu po e-shopu nezobrazily stejné notifikace znovu, protože byly stále označovány jako nové, přidal jsem do cookies seznam ID notifikací, které již proběhly, a ten se poté vyčistil při opakování cyklu AJAXu bez nové notifikace. Po zhotovení tohoto řešení jsem se rozhodl pokračovat v Push notifikacích, jelikož mé řešení pro obdržení notifikace v reálném čase vyžadovalo, aby byl koncový uživatel přítomný na e-shopu klienta.

```
$(document).ready(function loadnotify() {
    if (Notification.permission === "granted") {
        var blacklist = ExtractCookies('UsedNotifies')
        $.ajax({
            url: "/exec/notification.aspx?ids=" + blacklist,
        }).success(function (data) {
            data = data.split("|")
            if (data[0].length > 0) {
                var notifications = data[0].split(';');
                $.each(notifications, function (i) {
                    notifyMe(notifications[i]);
                });
            }
            if (data[1] != blacklist)
                InsertCookies('UsedNotifies', data[1], 0);
        })
        setTimeout(loadnotify, 60000);
    }
});
```

Výpis 4: Funkce pro kontrolu nově příchozích notifikací

Z nabídky třetích stran jsem si vybral Google Firebase, který sice vyžaduje přihlášení, ale funkci Push notifikací nemá zpoplatněnou. Při studování této varianty, mi kolega poskytl kostru svých Push notifikací, které ovšem využívají jiné služby, VAPID. Oproti Firebase mají výhodu, že se není potřeba nikam přihlašovat a třetí strana pouze pomocí Diffie-Hellmanova algoritmu poskytuje klíče od uživatelů, kteří povolují na svém prohlížeči pro danou stránku notifikace, a zajišťuje okamžité odeslání (pokud není nastaveno jinak) na uživatele se zvolenými klíči.

V administraci jsem rozšířil možnosti notifikací, které může klient využívat, o až tři tlačítka, ta kromě vlastních možností byla rozšířena o odkaz, na který měla po kliknutí přesměrovat. Jelikož

se mi nepovedlo naimplementovat funkci, kterou by si klient zpřístupnil nová notificační tlačítka podle vlastní potřeby, a zároveň aby se při znovuootevření již odeslané notifikace také zobrazil správný počet použitých tlačítek, musel jsem přidat nové přeražení v případě vynechaných tlačítek. Dále jsem administraci rozšířil o časové údaje, kdy se má zpráva začít zobrazovat (implicitně ihned) a kdy nejpozději, pokud měl odběratel zavřený prohlížeč (implicitně až 4 týdny), a nakonec o možnost vkládat do notifikace obrázky, těm byla přidána funkce vkládání pomocí manažeru obrázků, ale je podporována i možnost vložit URL adresu obrázku kdekoliv z internetu. O tuto možnost přišly odkazy, protože se notifikace odesílají přes service workery, které nepodporují některé funkce, ale v případě, že odběratel již má nějaký panel s odkazovanou stránkou, potom díky service workerům je možné neotvírat nový panel, ale pouze dát do popředí již otevřený, jelikož jsou schopny zjistit panely klientských stránek.

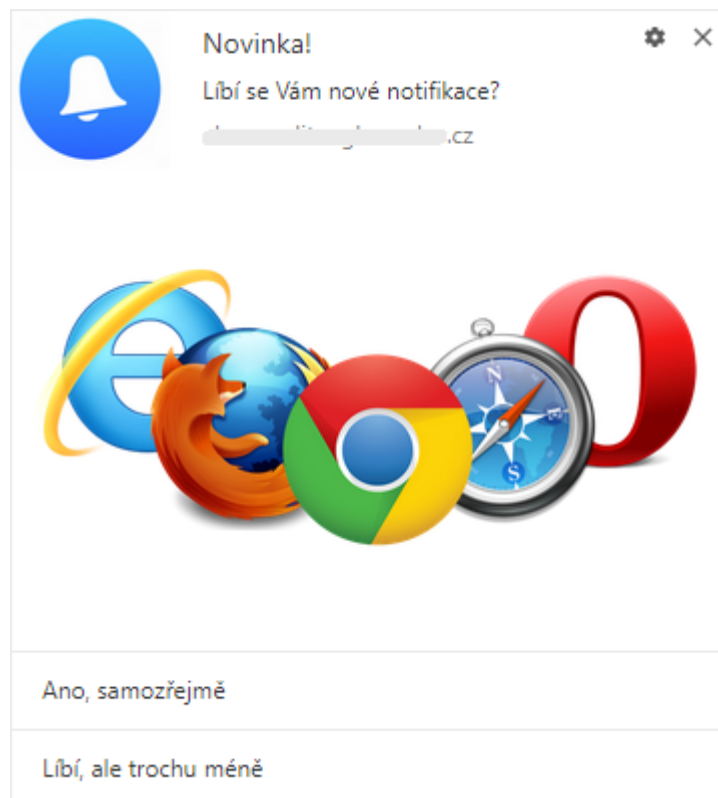
```
if (url.length != 0) {
  if (url.slice(-1) == '/') {
    url = url.slice(0, -1);
  }
  e.waitUntil(clients.matchAll({
    includeUncontrolled: true
  }).then(function (clientList) {
    for (var i = 0; i < clientList.length; i++) {
      var client = clientList[i];
      if (client.url == url || client.url == url + '/' && 'focus' in
        client)
        return client.focus();
    }
    return clients.openWindow(url);
  }));
}
```

Výpis 5: Hledání shody odkazu s otevřenými panely

Databázi jsem rozšířil o tyto nové možnosti a dále o nějaké statistické údaje, o tom kolik uživatelů notifikaci vidělo, kliklo na ni nebo kolik kliků zaznamenaly jednotlivá tlačítka. Přestože service workery nepodporují AJAX, zobrazují se tyto údaje zpět klientovi v reálném čase pomocí Fetch API [13]. Přidal jsem do databáze také údaje o klíších uživatelů, které VAPID poskytuje a navrhl systém identifikace s registrovanými uživateli, kteří nebyli přihlášení v době potvrzení odebrání notifikací. Po přihlášení se z XSLT šablony zavolá javascriptová funkce kontrolující odebrání notifikací a následně AJAX funkce, pomocí které se k záznamu s klíči prohlížeče zapíše ID registrovaného uživatele. Klient díky toho je schopen na základě existence ID u záznamu adresovat pouze své registrované zákazníky. Celé řešení je vytvořeno jako balíček, který se dá s pouze drobnými úpravami použít na dalších e-shopech.

Během vytváření notifikací vzniklo mnoho problémů, které zapříčinily několikanásobné přepisování již skoro hotového a hledání jiných alternativních řešení. Rozeslání notifikací na uživatelská zařízení bylo naimplementováno v C# narozdíl od administrace, která pracuje ve VBScript, proto bylo potřeba zavolat tento soubor bez přesměrování. Tuto operaci jsem chtěl provést pomocí funkce AJAX v klasickém javascriptu, ta ovšem z nějakého důvodu nefungovala. Při použití JQuery se sice ve většině implementací funkce AJAX zavolala nikoli však podle očekávání, vznikaly volání před ověřením integritních omezení, nebo vícečetná volání. Pomohl mi až kolega, který mi poradil použít interní funkci, která odešle požadavek na server a vrátí odpověď bez přesměrování, tato funkce již s ničím nekolidovala a fungovala dle potřeb.

Hlavní příčinou problémů byla často v současnosti slabá podpora notifikací, ale i HTML5 různými prohlížeči. Z testovaných prohlížečů dokázal zobrazit tlačítka pouze Chrome, a přestože samotné notifikace podporují mnoho tlačítek, ani u Chrome se nezobrazila více než dvě, proto bylo potřeba třetí tlačítko úplně odebrat. Původně použité vstupy datetime z HTML5 a formátování na ně bylo nutné rozdělit na date a time a znovu provést formátování. S datумы se poté objevil problém při vkládání resp. nevkládání do databáze. Datумы byly již formátovány v aplikaci dle standardu ISO, ale v databázi nereagovaly ani na přetypování, povedlo se mi je vypsat, až když jsem je začal vypisovat jako textový řetězec, ale ani ten jsem nebyl schopen v databázi konvertovat na čas. Využil jsem program SQL Profiler, abych zjistil jaká data se do procedury posílají, po zkopírování příkazu do SQL Server Management Studio se příkaz vždy vykonal správně a nedočkal jsem se žádné chybové hlášky, na kterou bych mohl reagovat. Začal jsem odesílat zvlášť datum i čas, abych zredukoval prostor, kde se může vyskytnout chyba a díky toho jsem odhalil, že samotný čas sice funguje v zápisu do DB, ale pro využití v praxi bude lepší část pro datum vyplnit dnešním datem, než ji nechat prázdnou. Jediná zbývající možnost, kde by se původní chyba mohla nalézat byla pomlčka/spojovník v datumech. Vzhledem k omezeným možnostem funkcí pro formátování času ve VBScript jsem raději řetězcovou funkcí nahradil znaky pomlček/spojovníků za lomítka a po konvertování v databázi se datum začalo správně ukládat. Následně jsem musel zakomentovat odložené odesílání, přestože se v dokumentaci uvádí, že by alespoň prohlížeč Chrome měl tuto vlastnost podporovat, byla ignorována i v několika různých formátech, stejně jako u ostatních prohlížečů. Pro potvrzení těchto skutečností, že není chyba v mé implementaci, jsem využil online generátor notifikací, se kterým jsem došel ke stejným výsledkům. Funkcím, které fungují částečně jsem přidal poznámku o omezeních pro klienta.



Obrázek 3: Přijatá notifikace v prohlížeči Chrome

Nepodařilo se mi obejít problém, který způsobuje chyba (popř. vlastnost) prohlížeče Chrome, kdy po submitu formuláře pro odeslání notifikace si klient může přechodem přes tuto stránku pomocí tlačítek prohlížeče pro zobrazení minulé nebo následné stránky nechtěně odeslat formulář znovu. Díky atributu tag, kterému nikdy nenechávám prázdnou hodnotu, se nemůže stát, že by byl koncový uživatel zaplaven notifikacemi, protože se zobrazuje vždy poslední notifikace se stejným atributem tag, ale pokud by uživatel již na notifikaci reagoval, zobrazila by se mu znovu, pokud uživatel nereagoval, výměnu notifikací by ani nepostřehl. K tomuto problému jsem našel několik tipů na úpravu cache, kterou si prohlížeč o stránce ukládá. Vyzkoušel jsem několik variant možností s pamětí cache s ověřením funkčnosti v prohlížeči Firefox, u prohlížeče Chrome bohužel beze změn. Pro účely testování jsem odeslal kolem 800 notifikací, z nichž drtivá většina následovala po větší, či menší úpravě kódu.

3.7.4 Zhodnocení

Použité technologie: JavaScript, jQuery, VBScript, C#, T-SQL, HTML, AJAX, XSLT

Časová náročnost: 20 dní

4 Využité znalosti a dovednosti

V průběhu mé individuální odborné praxe byly jako jedny z nejpoužívanějších technologií jazyky C# a T-SQL. Získané znalosti v předmětech *Programovací jazyky II* a *Vývoj informačních systémů* pro jazyk C# mi velmi pomohly, stejně jako předměty *Úvod do databázových systémů*, *Technologie databázových systémů I* a *Databázové a informační systémy* pro jazyk T-SQL popř. SQL. Velký přínos má také znalost vývojových prostředí Microsoft Visual Studio a SQL Server Management Studio nabytá na VŠB-TUO. A přestože mají ostatní jazyky vždy trochu odlišný syntax, mají určitý logický styl, ze kterého lze rovněž čerpat, proto pro mě měly nezanebatelný význam i jiné předměty s programovacími jazyky na bázi C a SQL, na vysoké i střední škole.

5 Chybějící znalosti a dovednosti

V prvních úkolech jsem se často dostával do kontaktu s technologií XSLT, kterou jsem do té doby neviděl, a citelně ovlivňovala trvání některých úkolů. Tu poté vystřídala technologie VBScript, kterou používají starší e-shopy ve své administrační části. VBScript alespoň logicky připomíná známé programovací jazyky, proto mi vždy stačilo jen upřesnit syntax. Poprvé jsem se také setkal s JavaScriptem popřípadě s jQuery a AJAX, přestože jsou na škole předměty, které se na tyto technologie zaměřují. Problémy mi také dělala komunikace mezi několika technologiemi, protože jsem se do té doby setkal s maximální kombinací dvou technologií C# a T-SQL v předmětech *Vývoj informačních systémů* a *Databázové a informační systémy*. V odborné praxi jsem se také setkal s novými situacemi, které škola může poskytnout jen s obtížemi, například práci na obrovském projektu z pohledu programu i databáze, zadávání úkolu i jeho odevzdání laické osobě a také práce v kolektivu.

6 Závěr

Absolvování individuální odborné praxe je cenná zkušenost a jsem velmi rád, že jsem si ji mohl vyzkoušet a díky ní se i profesně zlepšit. Mohl jsem si vyzkoušet pracovat v týmu lidí na reálných projektech, které používají spousty technologií. Práce na servisním oddělení zasahuje do profesí od databázisty, přes programátora, až po kodéra a díky tomu udržuje všeobecný přehled a pomáhají zjistit, která práce člověka nejvíce baví a čím by se chtěl dále zabývat. V průběhu této praxe jsem také řešil několik různorodých úkolů, jež jsem dopracoval do zdárného konce a několika lidem tak pomohl v jejich podnikání.

Firma NetDirect s.r.o. je velmi přátelský kolektiv a po mé zkušenosti bych tuto firmu doporučil každému, kdo má o tuto práci zájem, stejně tak bych doporučil absolvovat individuální odbornou praxi.

Literatura

- [1] NetDirect, [online], [cit. 2018-04-29], <https://www.netdirect.cz/>
- [2] C#, [online], [cit. 2018-04-29], <https://msdn.microsoft.com/cs-cz/library/dd492171.aspx>
- [3] T-SQL, [online], [cit. 2018-04-29], <https://docs.microsoft.com/en-us/sql/t-sql/language-reference?view=sql-server-2017>
- [4] ASP, [online], [cit. 2018-04-29], https://www.w3schools.com/asp/asp_introduction.asp
- [5] JavaScript, [online], [cit. 2018-04-29], <https://www.w3schools.com/js/default.asp>
- [6] XSLT, [online], [cit. 2018-04-29], https://www.w3schools.com/xml/xsl_intro.asp
- [7] XML, [online], [cit. 2018-04-29], <https://www.w3schools.com/xml/default.asp>
- [8] HTML, [online], [cit. 2018-04-29], <https://www.w3schools.com/html/default.asp>
- [9] CSS, [online], [cit. 2018-04-29], <https://www.w3schools.com/css/default.asp>
- [10] JQuery, [online], [cit. 2018-04-29], <https://www.w3schools.com/jquery/default.asp>
- [11] AJAX, [online], [cit. 2018-04-29], https://www.w3schools.com/js/js_ajax_intro.asp
- [12] SQL Search, [online], [cit. 2018-04-29], <https://www.red-gate.com/products/sql-development/sql-search/>
- [13] Fetch API, [online], [cit. 2018-04-29], <https://developers.google.com/web/updates/2015/03/introduction-to-fetch>